DOCUMENT RESUME

ED 149 713 IR 005 380

AUTHOR Bagley, Carole A.
TITLE CAI: VS CBE Languages. Authoring, How Soon?
PUB DATE Jun 74
NOTE 27p.; A Minnesota Educational Computing Consortium CAI Task Force research report

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS Administrator Attitudes; Classroom Environment; *Computer Assisted Instruction; *Computer Oriented Programs; Computer Programs; Cost Effectiveness; *Diffusion; Educational Innovation; Learning Theories; Programing Languages; Programing Problems; Teacher Attitudes
IDENTIFIERS BASIC Programing Language; *Computer Software

ABSTRACT

This paper gives a view of CAI (computer assisted instruction), computers in education, CAI author languages, and concepts for authoring. Distinctions are drawn among CAI, CBE (computer based education), CMI (computer managed instruction), and CGM (computer managed materials), and the functions of each are described. CAI has been slow in coming because of costs, educator and administrator conservatism, and a lack of adequate software and user program units. In choosing a CAI software package, one must consider hardware availability, BASIC capabilities, and the general applicability of author languages. Charts are provided which classify fourteen popular CAI author languages and systems according to forty capabilities outlined by C. Frye. The paper ends with a discussion of how to integrate the above factors with narrow learning theories in order to promote the use of CAI in the classroom. (DAG)

CAI: VS CBE LANGUAGES

AUTHORING HOW SOON?

by

Carole A. Bagley
Minnesota Educational Computing Consortium (MECC)
2520 Broadway Drive
Saint Paul, Minnesota 55113

CAI Task Force Research Report
for MECC in June, 1974

By the year 2000, computers are likely to surpass some of man's most human-like intellectual abilities, including perhaps some of his aesthetic and creative capacities, in addition to having some new kinds of capabilities that human beings do not have.

Eventually there will probably be computer consoles in every home, perhaps linked to public utility computers and permitting each user his private file space in a central computer, for uses such as consulting the Library of Congress, preparing inclome tax, keeping individual records, etc.[1]

Excerpt from The Year 2000

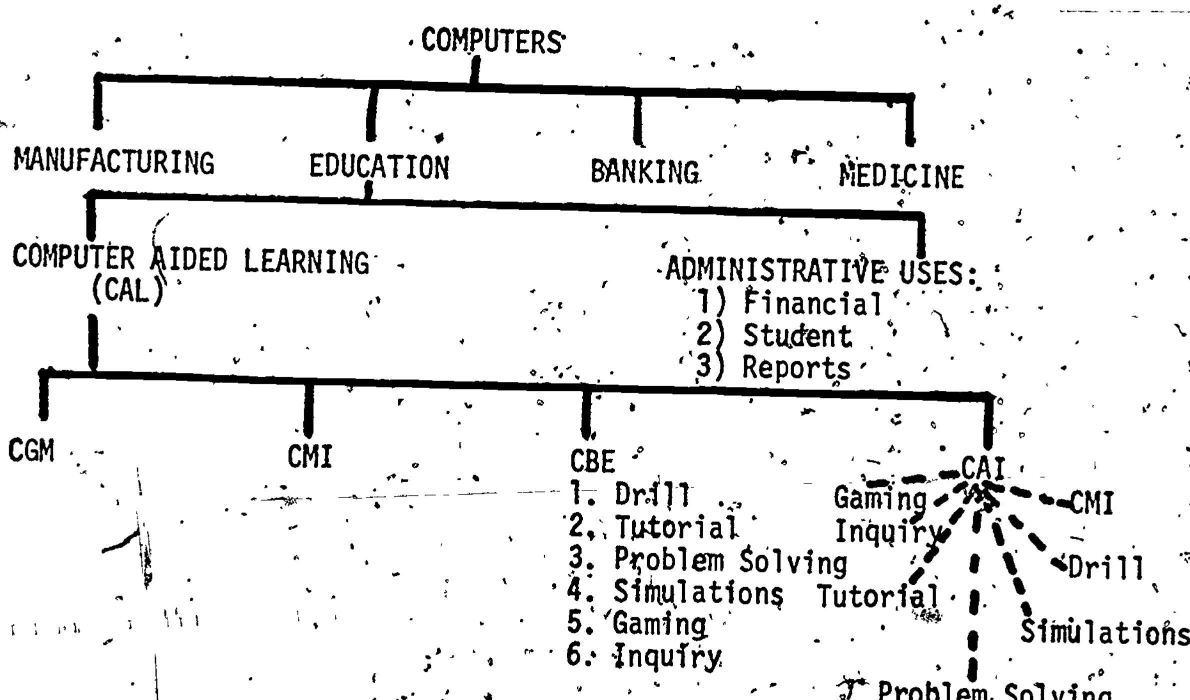
Computer Assisted Instruction (CAI) has taken on many definitions by such noted authors, educators and researchers as Richard Atkinson, Robert Glaser, A. Hickey, B. Hicks, S. Hunka, John Starkweather, H. Wilson and Karl Zinn. CAI is defined by some as a very broad category which includes any use of the computer in education , while others define it more specifically as instruction presented by the computer with no teacher intervention.

This paper will attempt to give the reader a detailed view of CAI, computers in education in general, CAI author languages and concepts for authoring.

Computers have become a part of every profession. They are used in manufacturing, medicine, banking and many other professions including education. The educational world uses the computer for administrative as well as for instructional purposes. Administratively, the computer is used for financial, student and report purposes. In the financial category, the computer helps to prepare budgets, accounts receivable, personnel payroll, accounts payable, inventory and assets. It is used for enrolling students, classroom scheduling, grade reporting, attendance and student programs. The computer is also used to produce administrative, financial, program cost, evaluation and planning reports.

Instructionally the computer is used for CAI (Computer Assisted Instruction), CBE (Computer Based Education), CMI (Computer Managed Instruction), and CGM (Computer Managed Materials). CAI used the computer to present instuction, CBE uses the computer as an alternative for instruction, CMI uses the computer to manage instruction and CGM uses the computer to generate materials for use in the classroom.

The following diagram show the relationship of CAI to computers in education in general:



A good CAI software system is capable of incorporating CMI within the lesson structure, whereas CBE and CGM are separate entities. CBE uses tutorials, problem solving, simulations, drill, inquiry and gaming singly as an aide for the classroom teacher. Units are used by teachers to supplement their everyday teaching. CBE is an alternative method used by teachers as a supplement to lecture, films and tours, e.g. a science class that is studying weather may use the program and unit called WEATHR so that the students may test their predictions for the next days weather against the computer's prediction.

In the tutorial computer role, the computer is a tutor. The student is taught a concept by the computer with no teacher interaction. All interaction is between the computer and the student. Two examples of the tutorial role are LATTTT and CLIMAT. LATTTT is a geograpny lesson which teaches the student about latitude whereas CLIMAT introduces the student to the climates of the world.

Drill and practice programs drill the student on his knowledge of a specific topic. A number of programs have been written such as CAICL and SPDRIL to test the students knowledge of addition, subtraction, multiplication and division of numbers. Other drill programs such as SPELL which drills an elementary or secondary student on subject related words and GERMAN which tests a student's knowledge of German verbs are in wide use throughout the state of Minnesota and by other such computers in education projects as Chicago; McComb, Mississippi; Philadelphia; New York and Oregon school districts.

The computer acts as a simulator for simulation programs where the student engages in a true to life situation. The student in many simulations must play a role and by entering certain values and making decisions he controls the simulation. ODELL1 simulates life in Lake Odell. The student plays the role of 1 out of 5 given fish and must make decisions as to whether he should try to attack another species or whether he should try to escape.

Gaming is competitive action between participants to achieve a goal. Many educators do not consider gaming to have any instructional merit, however, some actually do in that they teach the student that all subjects can be fun and interesting and actually have real applications. Such games as MARKET which allows two teams of students to manage the manufacturing, advertising and selling of a consumer article and BAGELS which teaches the logic of decision making are two examples of worthy games.

Problem solving programs fall into two categories, those which the student writes himself and those which are written by someone else but enable the student to solve another problem. In student-written programs, the student must define the problem logically and must write a program for the computer to understand. The computer will then do the necessary calculations and provide the answer to the student. Here the computer is an aide in teaching problem solving skills by programming. In the second category the computer is the problem solver. A student may wish to factor a certain trinomial so that he can solve a mathematical problem involving rectangles. The issue is not whether the student can factor, but whether he can solve a problem involving rectangles, therefore, factoring becomes a tedious task and why not use

the computer? A previously written program called FACTRI can be accessed at this point to factor as many trinomials as is supplied by the student.

The inquiry role of computer instruction means database storage, retrieval and analysis of information and also dialogue. Dialogue is virtually impossible with present technology because the need for computer software to analyze English sentence structure and/or human voice patterns cannot be met. Therefore, any further discussion of the inquiry role will not include dialogue. Inquiry programs such as INQUIR analyze large databases of election information, population statistics or other user-built databases and produce charts or other output for the student to study. GIS (Guidance Information System) also an inquiry program, reads large databases of 4 year colleges, 2 year colleges, scholarship information and vocational information and prints certain information for the user.

CGM programs generate a specified number of copies of a maze, puzzle drill exercise, etc. to be used in a classroom.

CMI programs monitor and guide students through a unit or course. The computer does not present course material or attempt to teach a concept. CMI programs guide a student through coursework by evaluating his performance on test given by the teacher. The scores are entered into the computer and a progress report is kept on each student. The student, depending upon complexity of the CMI program, may be told to read a certain book, work an exercise, take a test, or see the teacher for further work.

CAI program units use simulations, tutorials, drill, problem solving, gaming and inquiry together to present a concept. A program unit includes the program and all written documentation which supplements the program. The program teaches the concept by giving a pretest in the form of drill or problem solving, mainline instruction as tutorial, simulation, inquiry or gaming, remedial exercises and tutorial if needed and a posttest. If the author so chooses to keep progress reports on each student, to give specific instruction depending on student progress or to produce reports to help in teaching the class more effectively, this can in the more recent CAI software packages be incorporated into the CAI lesson structure.

Until just recently, CAI has usually been considered to be tutorial and drill and practice roles only. New develpments in CAI software systems have made it possible to write CAI program units which utilize all six roles of instruction: tutorial, drill, simulation, problem solving, gaming and inquiry and CMI together in one program unit if desired by the author.

CAI is definitely a plus for teachers, students and education, but it has been very slow in coming. Partially because of cost, but also due to educator and administrator conservatism and inadequate CAI system software and user program units.

The PLATO project has excelled in its hardware and software achievements, however, the cost of the PLATO terminal and the system itself is atrociously high. School districts do not have unlimited budgets, especially to spend in the area of computer instruction which has not proven itself to be an essential part of education.

The TICCIT project has brought the cost down somewhat, however, they have not begun to produce any user program units and therefore, cannot yet be made available.

The CRT terminal which is a necessity for CAI will come down in price as paper costs rise and as technology advances. Also, as educators become familiar with a CRT and compare it to the low speed, mechanical teletype, the cost will no longer be a factor.

Administrator and educator conservatism is another major issue which needs to be addressed. Many educators are afraid to change or cannot change because they fear they'll lose their jobs, they fear that a heavier workload will be forced upon them or they fear that the student will be ahead of them. CAI needs to be brought into all schools on a trial basis to show educators what is available to them and how it all works. There is really something better than the almighty teletype, but most educators don't realize it!!! CAI will not cause teachers to be layed off, but will probably cause an increased need for teachers. It is widely accepted fact that wherever a computer emerges more jobs are created. The role of the teacher will change, however, an increase in the workload will not occur. The teacher will no longer need to perform tedious duties such as grading and reporting and repetitious duties as lecture. Some lecture will still be necessary depending upon the teacher and the unit being studied, but much of the repetition that teachers despise will disappear.

Hicks and Hunka in their book The Teacher and The Computer devote a large section to the role of the teacher who uses CAI. "CAI definitely points to a new role for the teacher. This role involves working with teams of professionals and para-professionals, It will require the teacher to be more competent in terms of understanding the nature of his students and the subject matter. It will require a highly coordinated team effort, continuous evaluations, reviews and modifications in the CAI environment."[2]

A major problem with CAI as with any computer aided learning materials is the incompatibility of computer systems and therefore, the unavailability of CAI program units. Conversion between computer systems is a horrendous job. Hopefully, sometime in the near future, the ideal computer system will evolve which will alleviate the conversion problem. The system will be capable of translating any CAI program unit into its acceptable format. In addition to this feature, it will be cost effective, it will be extremely user oriented which means it will be easy to author materials, (no language to learn) and easy for a student to use, a database management capability will exist to store and retrieve large amounts of information quickly and efficiently for administrative as well as instructional purposes and it will include all necessary functional capabilities to provide good instruction.

A great deal of CBE development has been done by the Huntington Projects, Digital, NWREL, Hewlett Packard and other noted educational organizations. Some CMI programs have been developed and implemented by the Westinghouse Learning Center, CAM project and Florida State University. CAI program units have been developed by the PLATO project, University of Florida, University of California at Irvine and the University of Alberta CAI Laboratories. Various CAI languages including TUTOR, PLANIT, COURSEWRITER, APL and various versions of BASIC were used to create the CAI program units. Since these languages are specialized it is is improbable if not impossible to convert them to other computer systems.

The past few years, a large number of CBE, and some CMI and CAI programs have been implemented on computer systems around the United States. The cost and time spent on the conversions were extremely high. The cost and time spent on development and conversion of CAI program units could be reduced significantly on a large timesharing system which services CBE users and is equipped with a powerful, but user-oriented CAI language.

All CAI software packages provide the student with basically the same thing. Simply, the student is given text and is asked for an answer. The student never realizes he is being branched from frame to frame or that his answer is being matched according to a certain algorithm. Some CAI systems include only an author language and student interpreter, which may actually be the same code but could possibly be two separate programs, while others are highly sophisticated and include communications programs, data management routines, an author mode, interpretive routines and calculators. The former is referred to as a CAI author language, the latter to a CAI system. Some are easier to author, others have a large number of niceties as far as author language capabilities and hardware devices. A CAI system should contain enough basic qualities to at least make it worth the storage it takes up in the computer. The extra features can be added on after the basic features are chosen.

Zinn stated in his study Comparative Study of Languages for Programming Interactive Use of Computers in Instruction that "Education software is restricted by the nature of the languages developed and the available hardware on which software can be used. No present solution exists to the problem of need for appropriate author languages. Presently, therefore, there is no solution regarding sharability of prepared software. Computer assisted instruction development is inhibited by a need for compatible educational and computer software."[3]

When choosing a CAI software package, one should consider the following criteria:

1. Must run on available hardware.
2. Must have BASIC capabilities. Some criteria to consider are those developed by C. Frye:
   a. User Orientation
   b. Lesson Handling
   c. Record Handling - Files
   d. Conditional Branching
   e. Matching
   f. Calculation
   g. Communications [4]
3. The system should have general applicability. An author language which ties the author to a specialized format doesn't leave room for much creativity. A CAI package which can also be used for administrative purposes will be accepted more easily by educators and administrators.

A number of CAI software packages have been developed by educational institutions and computer manufacturers. "Some new languages have been motivated by deficiencies in old ones and emerge as only superficially different." [5] There are basically six levels of CAI author languages. The higher the number the easier the lesson is to author. Complete CAI software packages (more than just a CAI language) are intermixed within the six levels and some languages fall into more than one category.

1. Languages adopted from Compilers
   MENTOR
   CATO
   FAIL
   AUTHOR
2. Interpreters used for CAI and non-CAI applications
   BASIC
   APL
   LIDIA
3. Interpreter types developed specifically for CAI
   COURSEWRITER III
   NEWBASIC/CATALYST
   TUTOR
   PILOT
4. Frame-oriented interpretive languages
   PLANIT
   UIL/ASET
   SCHOLAR/TEACH
   IDF
5. Author prompted languages
   IDF
   SCHOLAR/TEACH
   PLANIT

6. Non-programmed fill in the screen type interpreters
   RPS-1100
   TICCIT

The chart on the following pages classifies a number of popular CAI author languages and systems according to the basic capabilities outlined by C. Frye. The Honeywell and IBM languages are not included because they use COURSEWRITER as a base and have added some extensions. The IBM interactive facility includes a special help provision for the student. It also includes the capability for a student to return to the point where he left off if he chooses to sign off. A monitor and supervisor mode have also been added. Honeywell has added a course manager routine and special routines for registering students.

LANGUAGE	ASET/UIL	APL	BASIC	NEW BASIC	COURSE-WRITER	IDF	LIDIA	PILOT	PLANIT	RPS	SCHOLAR/TEACH	TICCIT	TUTOR	MIL
I and II Message Switching	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes
Special Hardware	Yes	No	No	No	Yes	Yes	Yes	No	No?	Yes	Yes	Yes	Yes	Yes
Complex Algebraic Functions	No	Yes	Yes	Yes	No	No	No	No	Yes	Some	No	Some	No	Yes
Student can program while in CAI lesson	Yes	Yes	No	No	No	Yes	No	No (tutorials yes)	Yes?	No	No.	Some	No	Yes
Listing of student data 1-offline 2-online (reports)	1	No	1	1,2	1	2	1	No	1,2	2	1,2	1	1,2	1,2
Student can 1-modify 2-erase 3-insert entire words	1,2,3	1,2,3	No	No	No	No	No	?	No	1,2,3	No	1,2,3	No	No
Supports 1-simulation 2-problem solving 3-inquiry 4-text handling a-Separately b-Together in 1 program	1a,b 2a,b 3a,b 4a,b	1a,b 2a,b 4a,b	1a,b	1a,b	No	1a,b 2a,b	No	1a 2a	1a,b	1a,b 2a,b 3a,b 4a,b	1a,b 3a,b	1a,b 2a,b 3a,b 4a,b	1a,b 3a,b	1a;b 2a,b 3a,b (not good)
Lessons must be compiled before they can be used	Yes	Yes	No	No	Yes	No	Yes	No	No	No	No	No	Yes	Yes
Student can start where he left off after last log off.	No	No	No	No	No	Yes	No	No	Yes	Yes	No	Yes	No	Yes
Is a higher level language	No	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No	Yes	Yes

LANGUAGE	ASET/ UIL	APL	BASIC	NEW BASIC	COURSE-WRITER	IDF	LIDIA	PILOT	PLANIT	RPS	SCHOLAR/ TEACH	TICCIT	TUTOR	MIL
Author must know answer to calculate and include in answer list	No	No	No	No	No	Yes	No	No	No	No	No	No	No	No
CAI modes 1-authoring 2-execute 3-calculation 4-editing 5-1 mode which includes 1-4.	1,2,3 1	5 1	5 1	5 1	5 1	1,2, 3,4 2	? 1	1,2,4 1	5 2	1,2 1	5 2	1,2 2	1,2 1	1,2
III 1-author must code 2-automatic storing	No	No	No	No	Yes	Yes	No	No	No	Yes	No	Yes	Yes	1,2
Student may enter comments to the author	No	No	No	Yes	Yes	Yes	No	No	No	Yes	No	Yes	Yes	Yes
Good storage capability for information retrieval	?	No	No	No	No	No	No	No	No	Yes	Yes	Yes	?	?
Author can hold student answers temporarily for later output	No	No	No	No	No	No	No	Yes	No	Yes	No	No	No	Yes
IV branching to a Label	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	No	No	Yes	Yes
From 1 lesson to another No keeping track of reentry pt.	No	Usually No Some APL-Yes	No	No	No	No	Yes	No	Yes	Yes	No	Yes	No	Yes
To a subroutine & keeping track of reentry pt.	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Call Programs written in other languages	Yes	No	No ASM-Yes	No	No ASM-Yes	No	Yes	No	No	Yes	No	Yes	Yes	Yes

LANGUAGE	ASET/UIL	APL	BASIC	NEW BASIC	COURSE-WRITER	IDF	LIDIA	PILOT	PLANIT	RPS	SCHOLAR/TEACH	TICCIT	TUTOR	MIL
To current frame (I.E. +2)	No	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	?	Yes	No	Yes
To current status (I.E. Last output)	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes	?	No can branch last output	No	Yes
Can repeat a lesson depending on a condition	Yes	No	No	No	No	No	No	No	Yes	Yes	Yes	No	No	Yes
Branch depending on rating	No	No	No	No	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No
Conditional Branching due to content of a variable and/or flag of a student buffer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Implicit branching	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No
V Matching direct match	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Part word match	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Keyword match on 1 word	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Tolerance on numerics	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

LANGUAGE	ASET/UIL	APL	BASIC	NEW BASIC	COURSEWRITER	IDF	LIDIA	PILOT	PLANIT	RPS	SCHOLAR/TEACH	TICCIT	TUTOR	MIL
Algebraic match	Yes	No	No	No	No	No	No	No	Yes	No	Yes	Yes	No	Yes
Numeric	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Percentage	No	No	No	No	Yes	No	Yes	No	No	No	No	No	No	Yes
Selected Character string match	No	No	No	No	Yes	No	No	No	No	No	No	No	No	Yes
Consonant	No	No	No	No	No	No	Yes	No	No	No	No	No	No	No
Phonetic	No	No	No	No	No	No	Yes	No	Yes	No	No	No	No	No
Keyword on more than 1 word	No	No	No	No	No	No	Yes	No	Yes	No	No	No	Yes	Yes
Algebraically equivalent	No	No	No	No	No	No	No	No	No	No	No	No	No	No
VI calculator ability	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	No	?	No	Yes	Yes student
VII communication devices 1-teletype 2-CRT 3-audio 4-light pen 5-graphics 6-video tape 7-cassette tape 8-on line printer 9-graphics on terminal	1,2	1,2	1,2	1,2	1,2,3,4.5	1	1,2	1	1	2,7 8	1	2,3,6	2,3,9 slide projector not computer controlled	1,2

Extra Comments

NEWBASIC -- Reinforcement responses are stored in a file.
For loop is easier to understand (i.e. if pass> 3 then 850 instead of for I=1,3

```
          --------
          Next I
       850          )
```

can inquire a file and get back into program
Print command is nicer (i.e. 1000 Pr."----------LF
----------LF
----------CR

or

1000 frame ----------end of frame)

Keyword search can be executed to check for the key work only before designated string.

IDF - student and teacher must open files
student can use //Go to change course of study
author can use help command for authoring

PLATO - Uses full screen input
courseware is available in large quantity

PRS - can be used on line as an administrative system
uses full screen input

TICCIT - Uses full screen input
completely student controlled

All of the mentioned CAI systems or languages are fairly similar to the student. Message switching, student programming, calculator and student correction capabilities may vary between systems, but basically the student is provided information on a communication device and is asked at specific instances to provide a response. Some systems utilize full CRT screen capability and therefore give the student and author more room for expression.

The authoring capabilities, however, vary drastically. Some author languages require that a complex language and format be learned, others prompt the author when creating a lesson and the fill-in-the-screen type authoring provides successive CRT screens to the author to be filled out.

Following are authoring and/or student interaction examples of four of the CAI systems which were discussed.

NEWBASIC/CATALYST

Authoring

```
10 PR. "THIS IS A TRIVIAL EXAMPLE OF A TUTORIAL WHERE YOU MAY USE
@NBS TO RETRIEVE DATA FROM THE FILE/RIPLEY/. DO YOU HAVE YOUR FILE
INSTRUCTION SHEET WITH YOU?"
20 LET Y$=",YES, YUP, SURE, OF COURSE, AFFIRMATIVE,"
30 INPUT R$
40 IF IEQIV(R$,Y$,0) GO TO 70
50 PR. "PLEASE LOGOUT AND OBTAIN THE FILE INSTRUCTION SHEET. PRACTICE
USING IT ON A TERMINAL BEFORE TRYING THIS LESSON."
60 STOP
70 PR."HERE IS YOUR FIRST QUESTION.....
        NAME AN OBELISK FOUND IN AFRICA"
80 INPUT R$
90 IF ICO(R$,"KARNAK",1) CALL RRIN, GO TO 120
100 PR. "SORRY - YOUR ANSWER ISN'T ONE WE ANTICIPATED.
WE HAD THE 'TEMPLE OF KARNAK' IN MIND"
120 PR. "LET'S TRY ANOTHER QUESTION

 ........ETC....."
130 END
```

Student Interaction

```
RUN/TRIVIAL/
THIS IS A TRIVIAL EXAMPLE OF A TUTORIAL WHERE YOU MAY USE @NBS TO
RETRIEVE DATA FROM THE FILE/RIPLEY/. DO YOU HAVE YOUR FILE INSTRUCTION
SHEET WITH YOU?
?YUP
HERE IS YOUR FIRST QUESTION....
    NAME AN OBELISK FOUND IN AFRICA
?@NBS
VER. AUG 12 16:20
OPEN/RIPLEY/ FOR INPUT 2
```

```
INPUT FROM 2. A$(I) FOR I=1 TO 5
PRINT A$(I) FOR I=1 TO 5
4 ITEMS:
THE WASHINGTON MONUMENT
THE HONG KONG HILTON
THE TEMPLE OF KARNAK
THE HANGING GARDENS
CLOSE 2
EXIT
RESPOND TO LAST INPUT REQUEST
?THE TEMPLE OF KARNAK IS THE ANSWER
VERY GOOD INDEED!
LET'S TRY ANOTHER QUESTION

.....ETC.....
```

PILOT

Authoring

```
T:   HOW DO YOU THINK COMPUTERS CAN BE USED IN
:    EDUCATION?
*ANS A:
M:   MANAGE, ADMIN, INSTRUCT, PROGRAM, TEST
:    GRAD, COURSE, ANALYS, TEACH
TY:  YES, THAT'S A GOOD USE.  CAN YOU THINK OF ANY
:    OTHERS?
JY:  *ANS.
T:   DO YOU THINK THAT CHILDREN SHOULD HAVE AN
:    OPPORTUNITY TO LEARN ABOUT COMPUTERS?
A:
M:   NO
JY:  *NO
M:   TEACH
JY:  *TEACH
T:   ONE OF THE BEST WAYS IS TO GIVE THEM AN
:    OPPORTUNITY TO WRITE THEIR OWN PROGRAMS.
A:
M:   COMPUTER, ACCESS, COST
JY:  *COMP
*TEACH T:  OF COURSE THE TEACHERS SHOULD WRITE
:    PROGRAMS TOO, AS PART OF THE PRESENTATION
:    OF THE COURSE.  THE TEACHER SHOULDN'T HAVE
:    TO DEPEND ON DISTANT EXPERTS AND "PACKAGED
:    CURRICULUM" TO MAKE USE OF COMPUTER-
:    AIDED INSTRUCTION.  BESIDES, IN A DIALOGUE,
:    YOU HAVE TO KNOW THE STUDENT'S USE OF
:    LANGUAGE TO INTERPRET ANSWERS—THAT CAN
:    BE DIFFERENT FROM ONE GROUP OF CHILDREN
:    TO ANOTHER.
```

E:
*COMP T: WE'RE ASSUMING THAT MINICOMPUTERS
: ARE GOING TO BE CHEAP ENOUGH SO THAT
: SCHOOLS CAN AFFORD THEM, OR THAT REGIONAL
: EDUCATIONAL NETWORKS WILL BE DEVELOPED.
J: *TEACH
*NO T: REALLY? WHY DO YOU SAY THAT?
A:
J: *TEACH

PLANIT

Authoring

*Q

FRAME 2.00 LABEL=*MATH

2. SQ.
*LET'S SEE WHAT YOU REMEMBER ABOUT TEMPERATURE.
*USING F FOR DEGREES FAHRENHEIT AND C FOR DEGREES
*CENTIGRADE, WRITE THE FORMULA FOR CONVERTING FROM
*DEGREES FAHRENHEIT TO DEGREES CENTIGRADE.\
*HINT: F=9° C/5+32 CONVERTS FROM CENTIGRADE TO
FAHRENHEIT.
*
3. SA.
*O FORMULAS ON.
*A+c=(5/9)° (F-32)
*BF=9° c/5+32
*cc=(5/9)° F-32
*
4. SAT
*AF:NOW YOU'VE GOT IT.B:15
*BR:YOU'RE STILL CONVERTING FROM CENTIGRADE TO
FAHRENHEIT, TRY AGAIN...
*BC F:NOTE THE DIFFERENCE.C:B:OUT
*-R:

*-c:

IDF

Authoring

SECTION #1
OPTIONS? KEYWORD SEARCH
TEXT:
?GEORGE II RULED GREAT BRITAIN
?FROM 1727 TO 1760, AND DURING
?THAT REIGN HAD A SERIES OF WHIG
?PRIME MINISTERS
QUESTION:
?WHO WAS GEORGE II'S FIRST PRIME
?MINISTER:
CORRECT ANSWER GROUP:
?WALPOLE
REPLY FOR THIS GROUP:
?THAT'S RIGHT. ROBERT WALPOLE WAS
?PRIME MINISTER UNDER BOTH GEORGE
?I AND GEORGE II
WRONG ANSWER GROUP #1
?COMPTON
REPLY FOR THIS GROUP:
?NO, COMPTON WAS GEORGE II'S
?SECOND PRIME MINISTER. WHO WAS
?THE FIRST?
WRONG ANSWER GROUP #2
?WALP
?WOLP
REPLY FOR THIS GROUP:
?I THINK YOU HAVE THE RIGHT
?ANSWER, BUT THE WRONG SPELLING.
?TRY AGAIN. (ETC.)

Student Interaction

WHAT IS YOU ID NUMBER AND NAME?
1005, JOHNNY
JOHNNY SMITH? YES
COURSE NAME? HISTORY
GEORGE II RULED GREAT BRITAIN FROM
1727 TO 1760, AND DURING THAT
REIGN HAD A SERIES OF WHIG PRIME
MINISTERS.
WHO WAS GEORGE II'S FIRST PRIME
MINISTER? COMPTON, I THINK.
NO, COMPTON WAS GEORGE II'S
SECOND PRIME MINISTER. WHO WAS
THE FIRST? WAS IT WOLPINE?
I THINK YOU HAVE THE RIGHT ANSWER,
BUT THE WRONG SPELLING. TRY AGAIN.
I KNOW, IT WAS RICHARD WALPOLE
THAT'S RIGHT. ROBERT WALPOLE WAS
PRIME MINISTER UNDER BOTH GEORGE
I AND GEORGE II (ETC.)

Is it really necessary that the author language or authoring mode of a CAI system be easy to learn and use? Many CAI lesson authors and researchers argue that by using the authoring team approach which includes author, subject matter consultant and programmer, only the programmer will be involved with the actual coding. Since the programmer is familiar with programming technique and jargon there is no real need for an easy to learn, easy to use CAI language or author mode.

This may be true for large CAI authoring projects which have a large amount of funding, however, when smaller organizations attempt to reduce the authoring team to 1 or 2 people, the programmer is cut out first. Also, in large computer education projects where the staff is involved with servicing timeshare users and with development of computer applications, many teachers with little or no computer experience also become involved with authoring and require an easy-to-use author language or mode.

The authoring staff for a CAI project depends largely upon the financial situation of the organization. In a large project. each subject area could easily support a staff which includes 1 educational technologist who understands reinforcement. punishment and learning theories and has a feel for programmed learning and instructional computer systems, 1 subject matter consultant who is an expert in his subject area, 1 programmer and an advisory council made up of subject matter teachers. In a smaller organization, a pool of programmers who service all subject areas could be used as needed and subject matter consultants could be called in from consulting firms or school districts as needed.

CAI is capable of becoming a widely used and effective educational tool, however, it must first overcome a number of obstacles which are impeding its edvelopment. Many noted CAI authoring projects tend to concentrate on proving the educational worth of their material by stressing certain areas such as hardware or software capability or the people involved with the project. Two major CAI laboratories have repeatedly emphasized the hardware devices available to the user, the cost factor and the names involved with the direction of the project. They seem to evade some important issues which must be considered when developing and testing CAI program units.

Many of the previous CAI systems have not had the capabilities or have not really give enought thought or time to the following issues:

1) The need for better teaching strategies to utilize the capabilities of the computer. An advisory council should help to solve this problem.
2) The design of CAI programs to record student responses and to progress students through a lesson depending on these responses (CMI).
3) Using simulations, problem solving, storage and retrieval (inquiry) of large amounts of textual information together with tutorial and drill and practice.

4) The need for assessing learning behavior before and after CAI use.
5) Will a CRT-type terminal continue to attract students or will it lose its sparkle?
6) The need for a lower terminal cost by obtaining state funding or computer manufacturers price reduction.
7) Poor documentation of CAI units.
8) Convincing administrators and educators that CAI works and that it is worth the money which is being spent.
9) CAI cost versus traditional teaching cost and CBE cost.
10) The need for a CAI system to have the capability to branch from a CAI lesson to a program written in any language. This would be a step towards solving the problem of incompatible CAI systems.
11) CAI equipment and programming techniques rarely represent current technology.
12) Computer technology and education should jointly support CAI development, but rarely do so, partly because of a large communication gap between the two professions.

Documentation of CAI program units is perhaps the most critical point although each issue is extremely important and should be discussed before implementing CAI. Excellent documentation is necessary not only for the author to express his ideas for using the unit and to communicate the objectives and flow of the CAI unit.

CAI program units must be planned with great detail. Every alternative which can possibly occur must be carefully planned and written into the program. The computer when running a CAI program operates similarly to that of a teacher who executes his lesson plan precisely, with its alternate paths for individualized instruction. It is therefore necessary for the authoring team to document every move.

Although Bloom's Taxonomy of educational objectives is not completely adequate for developing CAI program units, it does describe student mental processes. The authoring team should consider the taxonomy when developing units so that some extremely challenging exercises are incorporated into the program unit along with those which are less challenging.

R.M. Gagne has attempted to set down a hierarchy of capabilities that can be established by learning. The CAI authoring team should also be aware of these learning capabilities so that the CAI program units do not deteriorate to using only the simplest approaches as memorization.

Gagne's 8 learning capabilities are defined as:

1) Response learning-response to a stimulus.
2) Identification or multiple discrimination learning—ability to distinguish a particular object from a variety of similar objects.
3) Chains or sequences—chaining response learning and discrimination learning to obtain more complex behavior.
4) Associations—association of the familiar with the unfamiliar so that unfamiliar pattern can be recalled later.
5) Concepts—classification of a variety of similar but not identical things into a category exemplifies concept formation.
6) Principles—given certain concepts, a rule evolves.
7) Problem solving—development of principles or rules for solving a class of problems.
8) Stategies—the manner in which a problem is solved.

Bloom's and Gagne's ideas on learning theory should certainly be kept on mind, however, these do not provide much help when attempting to define a CAI unit.

After considering all CAI issues, learning theories and CAI author language capabilities, the authoring team must plan for the development of a program unit. Student behavioral objectives should be defined first. Maturity level, ability factors, methods for evaluation, range of subject matter and specific topics to be included are to be outlined and documented as step 1 in the authoring process. After this initial step has been completed step 2 would be the development of an instructional strategy. Learning theory should be applied to the detailed design of the CAI unit at this time. The authoring team should decide what material will be used in a pretest situation, as mainline instruction, in remedial teaching and as a posttest. The team must also decide from the amount of subject matter to be included. How many sections of mainline instruction are needed. What should the student be able to do when he finishes the first mainline instruction sequence? How should the student exhibit what he has learned? How should the student be evaluated? What level of competence should be considered minimal? Each of these questions must be answered as part of the instuctional strategy. Each possible path must be considered and documented. Depending upon the availability of an advisory council, at this time it would be appropriate for the authoring team to consult the advisory council for any additions, corrections or oversights. After the advisory council members have approved the program unit, step 3 can begin. Step 3 is actual writing of the CAI lesson. At this time the detailed interaction with the student is written based upon the strategy which was outlined in step 2. Each interaction with the student must be considered and proper paths must be written so that the unit is indeed individualized. After the program unit has been completed step 4 can commence. Step 4 involves communicating the design to a programmer so that he can write the program to fit the authoring team's specifications.

Each of the four main steps in developing a CAI program unit should be documented for the user. The documentation should be as detailed as possible so that a user can see exactly what is happening when he uses the unit.

Good documentation is only one step towards promoting the use of CAI in the classroom. We need to keep plugging away—developing CAI units along with CBE materials, testing the units in the classroom and proving to the administrators and teachers that CAI is worthwhile. Some advantages which need to be stressed when discussing CAI with administrators and teachers are:

<u>For the Student</u>

1) Better and faster learning since the student is rated and advances according to his progress.

2) Student does not have to wait 2 or 3 days for tests to be corrected or papers handed back.

3) Personalized tutoring - every student answers, not just a few. Each student's responses are kept secret from other students.

<u>For the Teacher</u>

1) Takes away the drudgery of grading and creating drill material.

2) Gives more time for master teachers to write lessons for students.

3) No need for computer experience - easy to use

4) CAI keeps records

5) Teachers can receive progress reports

6) CAI is cost effective on a large timeshare system. It's the terminals that bring the costs so high.

Such fallacies as CAI makes the learning process too mechanical, more learning takes place with traditional instruction, CAI is another step towards depersonalizing instruction and CAI will take away teaching jobs have already been proven false, but when these comments are brought up over and over again by new people, they must be discussed and proven false again.

CAI program units, like any computer based instructional material, must be tested in the classroom before they are released. Such fallacies as those mentioned above can be proven false by including a questionnaire as part of the testing procedure. Some questions which should be considered are:

1) CAI makes learning too mechanical.

2) CAI is an inefficient use of my time.

3) I felt frustrated by the CAI environment.

4) Even interesting material would be boring when presented in a CAI manner.

5) I learned a great deal when using CAI and feel that I could learn more that with traditional instruction.

6) I prefer CAI to traditional methods of instruction.

7) CAI is another step towards depersonalizing education.

8) I learned more quickly with CAI.

9) CAI is more individualized to my needs.

10) I feel that I am boxed in front of this terminal with no human contact.

How soon will CAI be accepted by all educators, administrators, researchers and the general public? And even after it is accepted, how long will it take for CAI to be fully implemented in the classroom? Who can say?

Many obstacles must be overcome. By speaking in support of CAI and by fighting for the future of education, CAI will emerge, slowly, but it will emerge!

FOOTNOTES

1 Kahn, Herman et al., The Year 2000 (New York, 1967), p.135

2 Hicks, B.L. et al., The Teacher and the Computer (Philadelphia, 1972), p. 35

3 Zinn, Karl, "Comparative Study of Languages for Programming Interactive Use of Computers in Instruction," EDUCOM (September, 1969), p. 12

4 Frye, Charles, "CAI Languages: Capabilities and Applications," DATAMATION, (September, 1968), p. 15

5 Zinn, DATAMATION, p. 22

BIBLIOGRAPHY

Bushnell, Don: The Computer as an Instructional Tool: A summary, SDC(1964).

CDC: TUTOR Reference Manual, CDC(1974).

Dwyer, T.A.: Teacher-student authored CAI using the NEWBASIC/CATALYST System, U of Pittsburgh (1973).

Feingold, Samuel: PLANIT - A language From CAI, DATAMATION (1968).

Frye, Charles: CAI Languages: Capabilities and applications, DATAMATION (1968).

Gagne, R.M. and Lindval, C: The Implications of Instructional Objectives for Learning, U of Pittsburgh (1964).

Haggerty, N. and Latta, R.: Computer Assisted Instruction: Implications for the Northwest, Wester Washington State College (1971).

Hewlett, Packard: IDF Author manual. HP(1972)

Hickey, Albert: CAI: A Survey of the Literature, ENTELEK(1968).

Hicks, B.L. and Hunka, S.: The Teacher and The Computer, W.B. Saunders Co. (1972).

Honeywell: Author guide, Honeywell (1970).

Kahn, Herman and Wiener, Anthony: The Year 2000, A Framework For Speculation on the next 33 Years, McMillan and Company (1967)

Kanner, Joseph: CAI- The New Demonology?, DATAMATION(1968)

Luskin, et al. : Everytining you always want to know about CAI*, CUE (1972).

Mager, Robert: Preparing Instructional Objectives, Fearon Publishers (1962)

Mitre Corp: An overview of the TICCIT Program, Mitre Corp. (1974)

Renshaw, Kent and Simonson, Roger: CAI-Boon or Boondoggle?, DATAMATION (1974).

Rogers, James: Current problems in CAI, DATATMATION (1968).

Rubin, Sylvan: A simple Instructional language, Stanford Research Institute (1973).

BIBLIOGRAPHY

UNIVAC: ASET Description, UNIVAC(1973).

UNIVAC: RPS-1100 Users Guide, UNIVAC (1974).

Wallace, Scott: Bridging the gap between Teacher and Programmer, Hewlett Packard(1972).

Wilson, H.A. and Atkinson, Richard: Computer Assisted Instruction a book of Readings, Academic Press (1969).

Zinn, Karl: Comparative Study of Languages for Programming Interactive Use of Computers in Instuction, EDUCOM (1969).

Zinn, Karl: Instuctional Uses of Interactive Computer Systems, DATAMATION(1968).